Enstore DataBase, File Clerk and File Clerk Client modifications

Enstore database modifications for small files project:

file table:
packageId - character varying default '' - bfid of a package file, indicates that the file is a part of package and can not be accessed as an individual file directly on tape.

PackageFileCounter integer default -1 – counter of active files in package (gets updated only for a package file).

PackageFileNumber integer default 0 – total number of files in a package.

Package can be deleted when *PackageFileCounter=0.*

cache_status character varying default ''
archive_status character varying default ''
cache_mod_time timestamp without time zone
archive_mod_time timestamp without time zone
where:

- cache_status – shows if file is in cache/migration/purge status. It can have the following values:
  - created – file was written to cache
  - purging – file is being purged
  - purged – file was deleted in cache
  - cached – file is in cache.
- archive_status – file migration/purge status. It can have the following values:
  - archiving – file is being written to tape. This state is useful for the recovery from failure
  - archived – file was written to tape
  - staging – file is being staged to cache from tape
  - staged – file is in cache
- cache_mod_time – time when cache_status has changed
- archive_mod_time – time when archive_status has changed

New table will be created in file DB to hold a status of files in transition - f**iles_in_transition**

bfid character varying

file_status character varying default ''

cache_mod_time timestamp without time zone.

This table is needed to keep information about files which write or read requests involving cache were not yet completed.

File clerk modifications – details.

Existing File Clerk (FC) must be modified to notify Policy Engine (PE) Server about when files are created in cache and when read file is requested. The specific of enstore file write operation is such that the file may not get considered as written into enstore even if mover successfully completed data transfer. File is considered as written into enstore when enstore client (encp) sets the pnfsId of written file in enstore file database by calling a corresponding File Clerk Client (FCC) method, which in its turn send a message to FC. File Clerk will be modified to send CACHE_WRITTEN event to PE Server. A special process (thread) in File Clerk will set timeouts when File Clerk gets a new_bit_file request from disk mover. If set_pnfsid request does not arrive from encp for the corresponding bfId, File Clerk raises an alarm and makes an entry into the list of suspect files on disk. This is needed for the subsequent cleanup of not completed file writes. For read requests DM checks if requested file is on disk and if not it sends an open request to File Clerk, which sends CACHE_MISS event to PE Server.

Modifications to file clerk.

Existing method: new_bit_file.

1. Set cache_status=created
2. Set cache_mod_time=now
3. Put bfid, cache_status, cache_mod_time into files_in_transition table
4. Notify a timer process (thread) to check this entry in a specified time period

Time process (thread).

> If specified period expires and the corresponding entry is in the table send alarm that file transfer has not completed.

Existing method: set_pnfsid:

1. send CACHE_WRITTEN event to PE Server
2. set cache_status=cached
3. notify a timer process that no check for this bfId is needed
4. remove entry from files_in_transition table

New method: open_bitfile:

1. if cache_status==cached return success
2. otherwise set cache_status=staging
3. send CACHE_MISS event to PE Server
4. return cache_status (and other information in the ticket)

New method: set_cache_status:

modifies cache_status and archive_status

Input:

'bfid': bfid

'cache_status': legitimate_cache_status or None

'archive_status': legitimate_cache_status or None

If cache_status or archive_status is None – do not modify it.

Returns modified ticket.

New method: register_copies – creates entries in file_copies_map

input:

'bfid_list' – list of bfids of single files.

'map_bfid' – bfid to map the list of single file bfids to (aggregated file bfid).

ticket['bfid_list'][0] ---- ticket['map_bfid']

......

ticket['bfid_list'][N] ---- ticket['map_bfid']

modifies ticket['status']

sends ticket back

File Clerk Client Modifications

New method: open_bitfile.

- def open_bitfile(self, bf_id)

returns ticket returned by file clerk

New method: register_copies – counterpart to file clerk method

returns ticket returned by file clerk